

Doxygen_MPD_STAR_ONLINE_DB Reference Manual

Generated by Doxygen 1.3.7

Wed Sep 6 12:01:26 2006

Contents

1	Doxygen_MPD_STAR_ONLINE_DB Class Index	1
1.1	Doxygen_MPD_STAR_ONLINE_DB Class List	1
2	Doxygen_MPD_STAR_ONLINE_DB File Index	3
2.1	Doxygen_MPD_STAR_ONLINE_DB File List	3
3	Doxygen_MPD_STAR_ONLINE_DB Class Documentation	5
3.1	svtCoolingSender Class Reference	5
3.2	svtInterLocksSender Class Reference	16
3.3	svtRDOsSender Class Reference	26
4	Doxygen_MPD_STAR_ONLINE_DB File Documentation	39
4.1	svtCoolingDaemon.cc File Reference	39
4.2	svtCoolingSender.cc File Reference	40
4.3	svtCoolingSender.hh File Reference	41
4.4	svtCoolingSender_i.cc File Reference	42
4.5	svtInterLocksDaemon.cc File Reference	43
4.6	svtInterLocksSender.cc File Reference	44
4.7	svtInterLocksSender.hh File Reference	45
4.8	svtInterLocksSender_i.cc File Reference	46
4.9	svtRDOsDaemon.cc File Reference	47
4.10	svtRDOsSender.cc File Reference	48
4.11	svtRDOsSender.hh File Reference	49
4.12	svtRDOsSender_i.cc File Reference	50

Chapter 1

Doxygen_MPD_STAR_ONLINE_DB Class Index

1.1 Doxygen_MPD_STAR_ONLINE_DB Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

svtCoolingSender	5
svtInterLocksSender	16
svtRDOsSender	26

Chapter 2

Doxygen_MPD_STAR_ONLINE_DB File Index

2.1 Doxygen_MPD_STAR_ONLINE_DB File List

Here is a list of all files with brief descriptions:

svtCoolingDaemon.cc	39
svtCoolingSender.cc	40
svtCoolingSender.hh	41
svtCoolingSender_i.cc	42
svtInterLocksDaemon.cc	43
svtInterLocksSender.cc	44
svtInterLocksSender.hh	45
svtInterLocksSender_i.cc	46
svtRDOsDaemon.cc	47
svtRDOsSender.cc	48
svtRDOsSender.hh	49
svtRDOsSender_i.cc	50

Chapter 3

Doxygen_MPD_STAR_ONLINE_DB Class Documentation

3.1 svtCoolingSender Class Reference

```
#include <svtCoolingSender.hh>
```

Public Member Functions

- [svtCoolingSender](#) (const char *localDir)
- virtual [~svtCoolingSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- svtCooling [previousVals](#) [NUM_DB_ROWS]
- svtCooling [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- svtCooling [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [waterTempLimit](#)
ditto
- float [waterPressLimit](#)
- float [airTempLimit](#)

3.1.1 Constructor & Destructor Documentation

3.1.1.1 svtCoolingSender::svtCoolingSender (const char * *localDir*)

Definition at line 19 of file svtCoolingSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir); // note this ignores the sub dir tag
23     init("svtCooling"); // setup the file I/O
24     initDataBase();       // database connections
25     initTable();          // table definitions
26
27 }
```

3.1.1.2 virtual svtCoolingSender::~svtCoolingSender () [inline, virtual]

Definition at line 44 of file svtCoolingSender.hh.

```
44 {};
```

3.1.2 Member Function Documentation

3.1.2.1 bool svtCoolingSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 129 of file svtCoolingSender_i.cc.

```

129                                     {
130
131     svtCooling* pre=&previousVals[rowNumber];
132     svtCooling* cur=&tempVals[rowNumber];
133
134     if(fabs(pre->tempSE-cur->tempSE)>=waterTempLimit) return true;
```

```

135 if(fabs(pre->tempNE-cur->tempNE)>=waterTempLimit) return true;
136 if(fabs(pre->tempSW-cur->tempSW)>=waterTempLimit) return true;
137 if(fabs(pre->tempNW-cur->tempNW)>=waterTempLimit) return true;
138 if(fabs(pre->pressSE-cur->pressSE)>=waterPressLimit) return true;
139 if(fabs(pre->pressNE-cur->pressNE)>=waterPressLimit) return true;
140 if(fabs(pre->pressSW-cur->pressSW)>=waterPressLimit) return true;
141 if(fabs(pre->pressNW-cur->pressNW)>=waterPressLimit) return true;
142 if(fabs(pre->eastAirTemp-cur->eastAirTemp)>=airTempLimit) return true;
143 if(fabs(pre->westAirTemp-cur->westAirTemp)>=airTempLimit) return true;
144
145 /* example ... note -> change to any element requires db-update
146 * and thus returns true immediately
147 *
148 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
149 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
150 *
151 * ....
152 */
153
154 return false;
155 }

```

3.1.2.2 void svtCoolingSender::initDataBase () [virtual]

Definition at line 75 of file svtCoolingSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbSvt;
82
83 if( !( node = mgr->initConfig(dbT,dbD) ) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }

```

3.1.2.3 void svtCoolingSender::initQuery () [virtual]

Definition at line 53 of file svtCoolingSender_i.cc.

```

53                                     {
54 #define __METHOD__ "initQuery()"
55
56     ofstream to(queryFile);
57
58     if(!to.is_open()){
59         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
60         return;
61     }
62
63     to<<"SVT: GPIB_SE_Temperature" <<endl;
64     to<<"SVT: GPIB_NE_Temperature" <<endl;
65     to<<"SVT: GPIB_SW_Temperature" <<endl;
66     to<<"SVT: GPIB_SW_Temperature" <<endl;
67     to<<"SVT: GPIB_SE_Pressure" <<endl;
68     to<<"SVT: GPIB_NE_Pressure" <<endl;
69     to<<"SVT: GPIB_SW_Pressure" <<endl;

```

```

70     to<<"SVT:GPIB_NW_Pressure"<<endl;
71     to<<"SVT:GPIB_EastAir_Temp"<<endl;
72     to<<"SVT:GPIB_WestAir_Temp"<<endl;
73
74 /* example
75 *     for(int i=0;i<16;i++){
76 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
77 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
78 *         ....
79 *     }
80 * }
81 */
82
83     to.close();
84
85 #undef __METHOD__
86 }

```

3.1.2.4 void svtCoolingSender::initTable () [virtual]

Definition at line 30 of file svtCoolingSender.cc.

```

30     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("svtCooling")))
35         sendMess("Could not find table=svtCooling",dbmFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(svtCooling));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(svtCooling));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostringstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")"<<ends;
46         sendMess((ms.str()).c_str(),dbmFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table){
53         svtCooling* thv = (svtCooling*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(svtCooling));
55     }
56
57 #undef __METHOD__
58 };

```

3.1.2.5 void svtCoolingSender::initTags () [virtual]

Definition at line 66 of file svtCoolingSender.cc.

```

66     {
67 /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("svt");
70
71 }

```

3.1.2.6 bool svtCoolingSender::loadUserControls (const char * *name*, const char * *value*) [virtual]

Definition at line 20 of file svtCoolingSender_i.cc.

```

20
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31 if(strstr(name,"waterTempLimit")){
32     waterTempLimit=atof(value);
33     sendMess("waterTempLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
34     return true;
35 }
36 if(strstr(name,"waterPressLimit")){
37     waterPressLimit=atof(value);
38     sendMess("waterPressLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39     return true;
40 }
41 if(strstr(name,"airTempLimit")){
42     airTempLimit=atof(value);
43     sendMess("airTempLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
44     return true;
45 }
46
47 return false;
48 #undef __METHOD__
49 }

```

3.1.2.7 bool svtCoolingSender::nextLine (ifstream & *from*) [inline]

Definition at line 77 of file svtCoolingSender.hh.

```

77
78 if(!from.getline(mline,255))return false;
79 return true;
80 }

```

3.1.2.8 bool svtCoolingSender::queryData () [virtual]

Definition at line 91 of file svtCoolingSender.cc.

```

91
92 #define __METHOD__ "queryData()"
93
94 /*
95 * MORE THAN AN EXAMPLE...
96 * IF Standard SC-Query via "caGet" then,
97 * no need to change this method AT ALL
98 *
99 */
100

```

```

101 writeTime = (unsigned int)time(NULL); //for database write time
102
103 //char systemCmd[1024];
104 ostream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };

```

3.1.2.9 char * svtCoolingSender::readAny ()

Definition at line 197 of file svtCoolingSender.cc.

```

197                                     {
198
199     strcpy(tmpLine,mLine);
200     ptr1=tmpLine;
201     ptr2=strtok(ptr1," ");
202     if(!ptr2) return ptr2;
203     ptr2=strtok(NULL," ");
204     return ptr2;
205 }

```

3.1.2.10 bool svtCoolingSender::readData (ifstream & from) [virtual]

Definition at line 91 of file svtCoolingSender_i.cc.

```

91                                     {
92 #define __METHOD__ "readData(ifstream)"
93
94 mreadStatus=true;
95 memset(tempVals,0,NUM_DB_ROWS*sizeof(svtCooling));
96
97
98 int i=0;
99 char* c=__CLASS__;
100 char* m=__METHOD__;
101
102     if(!nextLine(from) || !readVal(tempVals[0].tempSE)) readError(__LINE__,c,m);
103     if(!nextLine(from) || !readVal(tempVals[0].tempNE)) readError(__LINE__,c,m);
104     if(!nextLine(from) || !readVal(tempVals[0].tempSW)) readError(__LINE__,c,m);
105     if(!nextLine(from) || !readVal(tempVals[0].tempNW)) readError(__LINE__,c,m);
106     if(!nextLine(from) || !readVal(tempVals[0].pressSE)) readError(__LINE__,c,m);
107     if(!nextLine(from) || !readVal(tempVals[0].pressNE)) readError(__LINE__,c,m);
108     if(!nextLine(from) || !readVal(tempVals[0].pressSW)) readError(__LINE__,c,m);
109     if(!nextLine(from) || !readVal(tempVals[0].pressNW)) readError(__LINE__,c,m);
110     if(!nextLine(from) || !readVal(tempVals[0].eastAirTemp)) readError(__LINE__,c,m);
111     if(!nextLine(from) || !readVal(tempVals[0].westAirTemp)) readError(__LINE__,c,m);
112
113 /* example format
114 *   for(int i=0;i<NUM_DB_ROWS;i++){
115 *   if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
116 *
117 *   ....
118 *
119 *   }
120 */

```

```
121
122   from.close();
123   return true;
124 #undef __METHOD__
125 }
```

3.1.2.11 bool svtCoolingSender::readData (const char * *fileName*) [virtual]

Definition at line 116 of file svtCoolingSender.cc.

```
116                                     {
117 #define __METHOD__ "readData(fileName)"
118
119   ifstream from(fileName);
120   if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122   return readData(from); // user implemented file read
123 #undef __METHOD__
124 }
```

3.1.2.12 void svtCoolingSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 82 of file svtCoolingSender.hh.

```
82                                     {
83   mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
84 }
```

3.1.2.13 bool svtCoolingSender::readVal (long long & *value*)

Definition at line 269 of file svtCoolingSender.cc.

```
269                                     {
270
271   if(!readAny())return false;
272   char* store[256];
273   value=strtoll(ptr2,store,10);
274   if(strlen(*store)>0) return false; // value is not a number
275
276   return true;
277 };
```

3.1.2.14 bool svtCoolingSender::readVal (long & *value*)

Definition at line 258 of file svtCoolingSender.cc.

```
258                                     {
259
260   if(!readAny())return false;
261
262   char* store[256];
263   value=strtoll(ptr2,store,10);
264   if(strlen(*store)>0) return false; // value is not a number
265
266   return true;
267 };
```

3.1.2.15 bool svtCoolingSender::readVal (int & value)

Definition at line 247 of file svtCoolingSender.cc.

```
247                                     {
248
249     if(!readAny()) return false;
250
251     char* store[256];
252     value=(int)strtol(ptr2,store,10);
253     if(strlen(*store)>0) return false; // value is not a number
254
255     return true;
256 };
```

3.1.2.16 bool svtCoolingSender::readVal (short & value)

Definition at line 236 of file svtCoolingSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.1.2.17 bool svtCoolingSender::readVal (double & value)

Definition at line 225 of file svtCoolingSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.1.2.18 bool svtCoolingSender::readVal (float & value)

Definition at line 214 of file svtCoolingSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.1.2.19 bool svtCoolingSender::readVal (char *& value)

Definition at line 207 of file svtCoolingSender.cc.

```

207                                     {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.1.2.20 bool svtCoolingSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file svtCoolingSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     svtCooling* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals     = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__METHOD__);
158
159     //char mess[256];
160     ostreamstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("svtCooling");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176     #undef __METHOD__
177 }
```

3.1.3 Member Data Documentation

3.1.3.1 float [svtCoolingSender::airTempLimit](#) [protected]

Definition at line 38 of file svtCoolingSender.hh.

3.1.3.2 int [svtCoolingSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file svtCoolingSender.hh.

3.1.3.3 char [svtCoolingSender::mline](#)[256] [protected]

Definition at line 29 of file svtCoolingSender.hh.

3.1.3.4 bool [svtCoolingSender::mreadStatus](#) [protected]

Definition at line 28 of file svtCoolingSender.hh.

3.1.3.5 svtCooling [svtCoolingSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file svtCoolingSender.hh.

3.1.3.6 char* [svtCoolingSender::ptr1](#) [protected]

Definition at line 31 of file svtCoolingSender.hh.

3.1.3.7 char * [svtCoolingSender::ptr2](#) [protected]

Definition at line 31 of file svtCoolingSender.hh.

3.1.3.8 svtCooling [svtCoolingSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file svtCoolingSender.hh.

3.1.3.9 char [svtCoolingSender::tmpline](#)[256] [protected]

Definition at line 30 of file svtCoolingSender.hh.

3.1.3.10 int [svtCoolingSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file svtCoolingSender.hh.

3.1.3.11 `svtCooling` `svtCoolingSender::updateVals[NUM_DB_ROWS]` [`protected`]

Definition at line 25 of file `svtCoolingSender.hh`.

3.1.3.12 `float` `svtCoolingSender::waterPressLimit` [`protected`]

Definition at line 37 of file `svtCoolingSender.hh`.

3.1.3.13 `float` `svtCoolingSender::waterTempLimit` [`protected`]

dito

Definition at line 36 of file `svtCoolingSender.hh`.

The documentation for this class was generated from the following files:

- [svtCoolingSender.hh](#)
- [svtCoolingSender.cc](#)
- [svtCoolingSender_i.cc](#)

3.2 svtInterLocksSender Class Reference

```
#include <svtInterLocksSender.h>
```

Public Member Functions

- [svtInterLocksSender](#) (const char *localDir)
ditto
- virtual [~svtInterLocksSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- svtInterLocks [previousVals](#) [NUM_DB_ROWS]
- svtInterLocks [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- svtInterLocks [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)

3.2.1 Constructor & Destructor Documentation

3.2.1.1 svtInterLocksSender::svtInterLocksSender (const char * *localDir*)

dito

Definition at line 18 of file svtInterLocksSender.cc.

```

18                                     {
19
20     initTags();
21     if(localDir) cd(localDir); // note this ignores the sub dir tag
22     init("svtInterLocks"); // setup the file I/O
23     initDataBase();          // database connections
24     initTable();             // table definitions
25
26 }
```

3.2.1.2 virtual svtInterLocksSender::~svtInterLocksSender () [inline, virtual]

Definition at line 41 of file svtInterLocksSender.hh.

```
41 {};
```

3.2.2 Member Function Documentation

3.2.2.1 bool svtInterLocksSender::hasChanged (int *rowNumber*) [virtual]

Definition at line 121 of file svtInterLocksSender_i.cc.

```

121                                     {
122
123     svtInterLocks* pre=&previousVals[rowNumber];
124     svtInterLocks* cur=&tempVals[rowNumber];
125
126     /* example ... note -> change to any element requires db-update
127     * and thus returns true immediately
128     *
129     *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
130     *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
131     *
132     * ....
133     */
134
135     return true;
136 }
```

3.2.2.2 void svtInterLocksSender::initDataBase () [virtual]

Definition at line 74 of file svtInterLocksSender.cc.

```

74                                     {
75     #define __METHOD__ "initDataBase()"
76
77     /* More than an example... swap user & dbTrg as per subsystem*/
78     mgr->setUser("stardb","");
```

```

79  StDbType   dbT = dbConditions;
80  StDbDomain dbD = dbSvt;
81
82  if( !( node = mgr->initConfig(dbT,dbD)) )
83      sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal, __LINE__, __CLASS__, __METHOD__);
84
85 #undef __METHOD__
86 }

```

3.2.2.3 void svtInterLocksSender::initQuery () [virtual]

Definition at line 38 of file svtInterLocksSender_i.cc.

```

38                                     {
39 #define __METHOD__ "initQuery()"
40
41     ofstream to(queryFile);
42
43     if(!to.is_open()){
44         sendMess("Open Failed ",queryFile,dbMFatal, __LINE__, __CLASS__, __METHOD__);
45         return;
46     }
47
48     to<<"SVT:SVTWaterStart"<<endl;
49     to<<"SVT:LVready"<<endl;
50     to<<"SVT:HVEastReady"<<endl;
51     to<<"SVT:HVWestReady"<<endl;
52     to<<"SVT:MischV_INT"<<endl;
53     to<<"SVT:Leak_INT"<<endl;
54     to<<"SVT:Global_INT"<<endl;
55     to<<"SVT:SVTWaterFlow_INT"<<endl;
56     to<<"SVT:SVTWaterTemp_INT"<<endl;
57     to<<"SVT:SVTWaterPress_INT"<<endl;
58     to<<"SVT:MiscSVT1_INT"<<endl;
59     to<<"SVT:MiscSVT2_INT"<<endl;
60     to<<"SVT:MiscSVT3_INT"<<endl;
61     to<<"SVT:MiscSVT4_INT"<<endl;
62
63 /* example
64 *     for(int i=0;i<16;i++){
65 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
66 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
67 *         ....
68 *     }
69 * */
70 */
71     to.close();
72
73 #undef __METHOD__
74 }

```

3.2.2.4 void svtInterLocksSender::initTable () [virtual]

Definition at line 29 of file svtInterLocksSender.cc.

```

29                                     {
30 #define __METHOD__ "initTable()"
31
32     StDbTable* table=0;
33     if(!(table=node->addDbTable("svtInterLocks")))

```

```

34     sendMess("Could not find table=svtInterLocks",dbMFatal,__LINE__,__CLASS__,__METHOD__);
35
36     memset(tempVals,0,NUM_DB_ROWS*sizeof(svtInterLocks));
37     memset(previousVals,0,NUM_DB_ROWS*sizeof(svtInterLocks));
38
39     int nrows;
40     int* elist = table->getElementID(nrows);
41     if(nrows!=NUM_DB_ROWS){
42         //char mess[256];
43         ostringstream ms;
44         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")"<<ends;
45         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
46     }
47     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
48
49     unsigned int timestamp=time(NULL);
50     mgr->setRequestTime(timestamp);
51     if(mgr->fetchDbTable(table)){
52         svtInterLocks* thv = (svtInterLocks*)table->GetTable();
53         memcpy(previousVals,thv,nrows*sizeof(svtInterLocks));
54     }
55
56 #undef __METHOD__
57 };

```

3.2.2.5 void svtInterLocksSender::initTags () [virtual]

Definition at line 65 of file svtInterLocksSender.cc.

```

65     {
66     /* more than an example -> swap "trg" to your subsys & add to email list*/
67     setEmailTo("porter@bnl.gov");
68     setDomainName("svt");
69
70 }

```

3.2.2.6 bool svtInterLocksSender::loadUserControls (const char * name, const char * value) [virtual]

Definition at line 20 of file svtInterLocksSender_i.cc.

```

20     {
21 #define __METHOD__ "loadUserControls(name,value)"
22
23 /* more than an example ... swap driftLimit to yours
24 * and duplicate this structure for each selection criteria
25 if(strstr(name,"driftLimit")){
26     driftLimit=atof(value);
27     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
28     return true;
29 }
30 */
31
32 return false;
33 #undef __METHOD__
34 }

```

3.2.2.7 `bool svtInterLocksSender::nextLine (ifstream & from)` [inline]

Definition at line 74 of file svtInterLocksSender.hh.

```

74                                     {
75   if(!from.getline(mline,255))return false;
76   return true;
77 }
```

3.2.2.8 `bool svtInterLocksSender::queryData ()` [virtual]

Definition at line 90 of file svtInterLocksSender.cc.

```

90                                     {
91 #define __METHOD__ "queryData()"
92
93 /*
94  * MORE THAN AN EXAMPLE...
95  * IF Standard SC-Query via "caGet" then,
96  * no need to change this method AT ALL
97  *
98  */
99
100 writeTime = (unsigned int)time(NULL); //for database write time
101
102 //char systemCmd[1024];
103 ostringstream scmd;
104 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
105
106 if(system((scmd.str()).c_str()))
107     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
108
109 return true;
110 #undef __METHOD__
111 };
```

3.2.2.9 `char * svtInterLocksSender::readAny ()`

Definition at line 196 of file svtInterLocksSender.cc.

```

196                                     {
197
198   strcpy(tmpline,mline);
199   ptr1=tmpline;
200   ptr2=strtok(ptr1," ");
201   if(!ptr2) return ptr2;
202   ptr2=strtok(NULL," ");
203   return ptr2;
204 }
```

3.2.2.10 `bool svtInterLocksSender::readData (ifstream & from)` [virtual]

Definition at line 80 of file svtInterLocksSender_i.cc.

```

80                                     {
81 #define __METHOD__ "readData(ifstream)"
```

```

82
83 mreadStatus=true;
84 memset(tempVals,0,NUM_DB_ROWS*sizeof(svtInterLocks));
85
86
87 int i=0;
88 char* c=__CLASS__;
89 char* m=__METHOD__;
90 if(!nextLine(from) || !readVal(tempVals[0].waterStart)) readError(__LINE__,c,m);
91 if(!nextLine(from) || !readVal(tempVals[0].lvReady)) readError(__LINE__,c,m);
92 if(!nextLine(from) || !readVal(tempVals[0].hvEastReady)) readError(__LINE__,c,m);
93 if(!nextLine(from) || !readVal(tempVals[0].hvWestReady)) readError(__LINE__,c,m);
94 if(!nextLine(from) || !readVal(tempVals[0].miscHv)) readError(__LINE__,c,m);
95 if(!nextLine(from) || !readVal(tempVals[0].leak)) readError(__LINE__,c,m);
96 if(!nextLine(from) || !readVal(tempVals[0].global)) readError(__LINE__,c,m);
97 if(!nextLine(from) || !readVal(tempVals[0].waterFlow)) readError(__LINE__,c,m);
98 if(!nextLine(from) || !readVal(tempVals[0].waterTemp)) readError(__LINE__,c,m);
99 if(!nextLine(from) || !readVal(tempVals[0].waterPress)) readError(__LINE__,c,m);
100 if(!nextLine(from) || !readVal(tempVals[0].miscSvt1)) readError(__LINE__,c,m);
101 if(!nextLine(from) || !readVal(tempVals[0].miscSvt2)) readError(__LINE__,c,m);
102 if(!nextLine(from) || !readVal(tempVals[0].miscSvt3)) readError(__LINE__,c,m);
103 if(!nextLine(from) || !readVal(tempVals[0].miscSvt4)) readError(__LINE__,c,m);
104
105 /* example format
106 * for(int i=0;i<NUM_DB_ROWS;i++){
107 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(l,c,m);
108 *
109 *     ....
110 *
111 * }
112 */
113
114 from.close();
115 return true;
116 #undef __METHOD__
117 }

```

3.2.2.11 bool svtInterLocksSender::readData (const char * *fileName*) [virtual]

Definition at line 115 of file svtInterLocksSender.cc.

```

115
116 #define __METHOD__ "readData(fileName)"
117
118 ifstream from(fileName);
119 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
120
121 return readData(from); // user implemented file read
122 #undef __METHOD__
123 }

```

3.2.2.12 void svtInterLocksSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 79 of file svtInterLocksSender.hh.

```

79
80 mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
81 }

```

3.2.2.13 bool svtInterLocksSender::readVal (long long & value)

Definition at line 268 of file svtInterLocksSender.cc.

```
268                                     {
269
270     if(!readAny())return false;
271     char* store[256];
272     value=strtoll(ptr2,store,10);
273     if(strlen(*store)>0) return false; // value is not a number
274
275     return true;
276 };
```

3.2.2.14 bool svtInterLocksSender::readVal (long & value)

Definition at line 257 of file svtInterLocksSender.cc.

```
257                                     {
258
259     if(!readAny())return false;
260
261     char* store[256];
262     value=strotol(ptr2,store,10);
263     if(strlen(*store)>0) return false; // value is not a number
264
265     return true;
266 };
```

3.2.2.15 bool svtInterLocksSender::readVal (int & value)

Definition at line 246 of file svtInterLocksSender.cc.

```
246                                     {
247
248     if(!readAny()) return false;
249
250     char* store[256];
251     value=(int)strotol(ptr2,store,10);
252     if(strlen(*store)>0) return false; // value is not a number
253
254     return true;
255 };
```

3.2.2.16 bool svtInterLocksSender::readVal (short & value)

Definition at line 235 of file svtInterLocksSender.cc.

```
235                                     {
236
237     if(!readAny()) return false;
238
239     char* store[256];
240     value=(short)strotol(ptr2,store,10);
241     if(strlen(*store)>0) return false; // value is not a number
242
243     return true;
244 };
```

3.2.2.17 bool svtInterLocksSender::readVal (double & value)

Definition at line 224 of file svtInterLocksSender.cc.

```
224                                     {
225
226   if(!readAny())return false;
227
228   char* store[256];
229   value=strtod(ptr2,store);
230   if(strlen(*store)>0) return false; // value is not a number
231
232   return true;
233 };
```

3.2.2.18 bool svtInterLocksSender::readVal (float & value)

Definition at line 213 of file svtInterLocksSender.cc.

```
213                                     {
214
215   if(!readAny()) return false;
216
217   char* store[256];
218   value=(float)strtod(ptr2,store);
219   if(strlen(*store)>0) return false; // value is not a number
220
221   return true;
222 };
```

3.2.2.19 bool svtInterLocksSender::readVal (char *& value)

Definition at line 206 of file svtInterLocksSender.cc.

```
206                                     {
207
208   if(!readAny()) return false;
209   strcpy(value,ptr2);
210   return true;
211 }
```

3.2.2.20 bool svtInterLocksSender::updateDb (const char *fileName) [virtual]

Definition at line 126 of file svtInterLocksSender.cc.

```
126                                     {
127 #define __METHOD__ "updateDb(filename)"
128
129   if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
130
131   int* elements;
132   svtInterLocks* vals;
133   int numRows = 0;
134
135   if(writeRequired()){
136
```

```

137     numRows=NUM_DB_ROWS;
138     elements=elementList;
139     vals = tempVals;
140
141 } else {
142
143     for(int i=0; i<NUM_DB_ROWS; i++){
144         if(hasChanged(i)){
145             updateElements[numRows]=elementList[i];
146             updateVals[numRows] = tempVals[i];
147             previousVals[i]=tempVals[i];
148             numRows++;
149         }
150     }
151
152     elements = updateElements;
153     vals     = updateVals;
154 }
155
156 if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__MET
157
158 //char mess[256];
159 ostreamstream sn;
160 sn<<"Will Update " <<numRows<<" of " <<NUM_DB_ROWS<<" rows " <<ends;
161 sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
162
163 StDbTable* dbTable=node->findTable("svtInterLocks");
164 dbTable->SetTable((char*)vals, numRows, elements);
165 mgr->setStoreTime(writeTime);
166
167 if(!mgr->storeDbTable(dbTable)) {
168     addBackLog(writeTime);
169     return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
170 }
171
172 if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
173
174 return true;
175 #undef __METHOD__
176 }

```

3.2.3 Member Data Documentation

3.2.3.1 `int svtInterLocksSender::elementList[NUM_DB_ROWS]` [protected]

Definition at line 24 of file `svtInterLocksSender.hh`.

3.2.3.2 `char svtInterLocksSender::mline[256]` [protected]

Definition at line 29 of file `svtInterLocksSender.hh`.

3.2.3.3 `bool svtInterLocksSender::mreadStatus` [protected]

Definition at line 28 of file `svtInterLocksSender.hh`.

3.2.3.4 `svtInterLocks svtInterLocksSender::previousVals[NUM_DB_ROWS]` [protected]

Definition at line 22 of file `svtInterLocksSender.hh`.

3.2.3.5 `char* svtInterLocksSender::ptr1` [protected]

Definition at line 31 of file svtInterLocksSender.hh.

3.2.3.6 `char * svtInterLocksSender::ptr2` [protected]

Definition at line 31 of file svtInterLocksSender.hh.

3.2.3.7 `svtInterLocks svtInterLocksSender::tempVals[NUM_DB_ROWS]` [protected]

Definition at line 23 of file svtInterLocksSender.hh.

3.2.3.8 `char svtInterLocksSender::tmpline[256]` [protected]

Definition at line 30 of file svtInterLocksSender.hh.

3.2.3.9 `int svtInterLocksSender::updateElements[NUM_DB_ROWS]` [protected]

Definition at line 26 of file svtInterLocksSender.hh.

3.2.3.10 `svtInterLocks svtInterLocksSender::updateVals[NUM_DB_ROWS]` [protected]

Definition at line 25 of file svtInterLocksSender.hh.

The documentation for this class was generated from the following files:

- [svtInterLocksSender.hh](#)
- [svtInterLocksSender.cc](#)
- [svtInterLocksSender_i.cc](#)

3.3 svtRDOsSender Class Reference

```
#include <svtRDOsSender.hh>
```

Public Member Functions

- [svtRDOsSender](#) (const char *localDir)
- virtual [~svtRDOsSender](#) ()
- virtual void [initTable](#) ()
- virtual void [initTags](#) ()
- virtual void [initDataBase](#) ()
- virtual bool [loadUserControls](#) (const char *name, const char *value)
- virtual void [initQuery](#) ()
- virtual bool [queryData](#) ()
- virtual bool [readData](#) (const char *fileName)
- virtual bool [updateDb](#) (const char *fileName)
- virtual bool [readData](#) (ifstream &from)
- virtual bool [hasChanged](#) (int rowNumber)
- char * [readAny](#) ()
- bool [readVal](#) (char *&value)
- bool [readVal](#) (float &value)
- bool [readVal](#) (double &value)
- bool [readVal](#) (short &value)
- bool [readVal](#) (int &value)
- bool [readVal](#) (long &value)
- bool [readVal](#) (long long &value)
- bool [nextLine](#) (ifstream &from)
- void [readError](#) (int l, char *c, char *m)

Protected Attributes

- svtRDOs [previousVals](#) [NUM_DB_ROWS]
- svtRDOs [tempVals](#) [NUM_DB_ROWS]
- int [elementList](#) [NUM_DB_ROWS]
- svtRDOs [updateVals](#) [NUM_DB_ROWS]
- int [updateElements](#) [NUM_DB_ROWS]
- bool [mreadStatus](#)
- char [mline](#) [256]
- char [tmpline](#) [256]
- char * [ptr1](#)
- char * [ptr2](#)
- float [vdriftLimit](#)
ditto
- float [tdriftLimit](#)
- float [cdriftLimit](#)

3.3.1 Constructor & Destructor Documentation

3.3.1.1 svtRDOsSender::svtRDOsSender (const char * localDir)

Definition at line 19 of file svtRDOsSender.cc.

```

19                                     {
20
21     initTags();
22     if(localDir) cd(localDir);// note this ignores the sub dir tag
23     init("svtRDOs"); // setup the file I/O
24     initDataBase();      // database connections
25     initTable();         // table definitions
26
27 }
```

3.3.1.2 virtual svtRDOsSender::~~svtRDOsSender () [inline, virtual]

Definition at line 44 of file svtRDOsSender.hh.

```
44 {};
```

3.3.2 Member Function Documentation

3.3.2.1 bool svtRDOsSender::hasChanged (int rowNumber) [virtual]

Definition at line 192 of file svtRDOsSender_i.cc.

```

192                                     {
193
194     svtRDOs* pre=&previousVals[rowNumber];
195     svtRDOs* cur=&tempVals[rowNumber];
196     if(fabs(pre->v1-cur->v1)>=vdriftLimit) return true;
197     if(fabs(pre->v2-cur->v2)>=vdriftLimit) return true;
198     if(fabs(pre->v3-cur->v3)>=vdriftLimit) return true;
199     if(fabs(pre->p6-cur->p6)>=vdriftLimit) return true;
200     if(fabs(pre->mv-cur->mv)>=vdriftLimit) return true;
201     if(fabs(pre->m6-cur->m6)>=vdriftLimit) return true;
202     if(fabs(pre->v1Curr-cur->v1Curr)>=cdriftLimit) return true;
203     if(fabs(pre->v2Curr-cur->v2Curr)>=cdriftLimit) return true;
204     if(fabs(pre->v3Curr-cur->v3Curr)>=cdriftLimit) return true;
205     if(fabs(pre->p6Curr-cur->p6Curr)>=cdriftLimit) return true;
206     if(fabs(pre->mvCurr-cur->mvCurr)>=cdriftLimit) return true;
207
208     if(fabs(pre->northTemp-cur->northTemp)>=tdriftLimit) return true;
209     if(fabs(pre->southTemp-cur->southTemp)>=tdriftLimit) return true;
210     if(fabs(pre->hvBoardTemp-cur->hvBoardTemp)>=tdriftLimit) return true;
211     if(fabs(pre->optTemp-cur->optTemp)>=tdriftLimit) return true;
212
213     if(fabs(pre->lgRingCurr-cur->lgRingCurr)>=cdriftLimit) return true;
214     if(fabs(pre->rgRingCurr-cur->rgRingCurr)>=cdriftLimit) return true;
215     if(fabs(pre->lgAnodeCurr-cur->lgAnodeCurr)>=cdriftLimit) return true;
216     if(fabs(pre->rgAnodeCurr-cur->rgAnodeCurr)>=cdriftLimit) return true;
217
218     if(fabs(pre->hvVolt-cur->hvVolt)>=vdriftLimit) return true;
219     if(fabs(pre->hvCurr-cur->hvCurr)>=cdriftLimit) return true;
220
221     if(strcmp(pre->hvTrip,cur->hvTrip) != 0) return true;
222     if(pre->lvFault!=cur->lvFault) return true;
```

```

223
224 /* example ... note -> change to any element requires db-update
225 * and thus returns true immediately
226 *
227 *if(fabs(pre->ch0Voltage-cur->ch0Voltage)>=driftLimit) return true;
228 *if(fabs(pre->ch1Voltage-cur->ch1Voltage)>=driftLimit) return true;
229 *
230 * ....
231 */
232
233 return false;
234 }

```

3.3.2.2 void svtRDOsSender::initDataBase () [virtual]

Definition at line 75 of file svtRDOsSender.cc.

```

75                                     {
76 #define __METHOD__ "initDataBase()"
77
78 /* More than an example... swap user & dbTrg as per subsystem*/
79 mgr->setUser("stardb","");
80 StDbType dbT = dbConditions;
81 StDbDomain dbD = dbSvt;
82
83 if( !( node = mgr->initConfig(dbT,dbD) ) )
84     sendMess("Connect Failed ",mgr->printDbName(dbT,dbD),dbMFatal,__LINE__,__CLASS__,__METHOD__);
85
86 #undef __METHOD__
87 }

```

3.3.2.3 void svtRDOsSender::initQuery () [virtual]

Definition at line 58 of file svtRDOsSender_i.cc.

```

58                                     {
59 #define __METHOD__ "initQuery()"
60
61     ofstream to(queryFile);
62
63     if(!to.is_open()){
64         sendMess("Open Failed ",queryFile,dbMFatal,__LINE__,__CLASS__,__METHOD__);
65         return;
66     }
67
68     for(int i=0; i<72; i++){
69         to<<"SVT:P"<<port[i]<<"_V1_RDO"<<RDO[i]<<endl;
70         to<<"SVT:P"<<port[i]<<"_V2_RDO"<<RDO[i]<<endl;
71         to<<"SVT:P"<<port[i]<<"_V3_RDO"<<RDO[i]<<endl;
72         to<<"SVT:P"<<port[i]<<"_P6_RDO"<<RDO[i]<<endl;
73         to<<"SVT:P"<<port[i]<<"_MV_RDO"<<RDO[i]<<endl;
74         to<<"SVT:P"<<port[i]<<"_M6_RDO"<<RDO[i]<<endl;
75         to<<"SVT:P"<<port[i]<<"_V1_Curr_RDO"<<RDO[i]<<endl;
76         to<<"SVT:P"<<port[i]<<"_V2_Curr_RDO"<<RDO[i]<<endl;
77         to<<"SVT:P"<<port[i]<<"_V3_Curr_RDO"<<RDO[i]<<endl;
78         to<<"SVT:P"<<port[i]<<"_P6_Curr_RDO"<<RDO[i]<<endl;
79         to<<"SVT:P"<<port[i]<<"_MV_Curr_RDO"<<RDO[i]<<endl;
80         to<<"SVT:P"<<port[i]<<"_T1_RDO"<<RDO[i]<<endl;
81         to<<"SVT:P"<<port[i]<<"_T2_RDO"<<RDO[i]<<endl;
82         to<<"SVT:P"<<port[i]<<"_T3_RDO"<<RDO[i]<<endl;
83         to<<"SVT:P"<<port[i]<<"_T4_RDO"<<RDO[i]<<endl;

```

```

84     to<<"SVT:P"<<port[i]<<"_G1_RDO"<<RDO[i]<<endl;
85     to<<"SVT:P"<<port[i]<<"_G2_RDO"<<RDO[i]<<endl;
86     to<<"SVT:P"<<port[i]<<"_G3_RDO"<<RDO[i]<<endl;
87     to<<"SVT:P"<<port[i]<<"_G4_RDO"<<RDO[i]<<endl;
88     to<<"SVT:HV_RD_MV_P"<<port[i]<<"_RDO"<<RDO[i]<<endl;
89     to<<"SVT:HV_RD_MC_P"<<port[i]<<"_RDO"<<RDO[i]<<endl;
90     to<<"SVT:HV_HVTripped_P"<<port[i]<<"_RDO"<<RDO[i]<<endl;
91     to<<"SVT:HV_LVFault_P"<<port[i]<<"_RDO"<<RDO[i]<<endl;
92     }
93 /* example
94 *     for(int i=0;i<16;i++){
95 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".E"<<endl;
96 *         to<<"TRGhv:SUB_RD_V_1:"<<i<<".F"<<endl;
97 *         ....
98 *     }
99 * */
100 */
101
102     to.close();
103
104 #undef __METHOD__
105 }

```

3.3.2.4 void svtRDOsSender::initTable () [virtual]

Definition at line 30 of file svtRDOsSender.cc.

```

30     {
31 #define __METHOD__ "initTable()"
32
33     StDbTable* table=0;
34     if(!(table=node->addDbTable("svtRDOs")))
35         sendMess("Could not find table=svtRDOs",dbMFatal,__LINE__,__CLASS__,__METHOD__);
36
37     memset(tempVals,0,NUM_DB_ROWS*sizeof(svtRDOs));
38     memset(previousVals,0,NUM_DB_ROWS*sizeof(svtRDOs));
39
40     int nrows;
41     int* elist = table->getElementID(nrows);
42     if(nrows!=NUM_DB_ROWS){
43         //char mess[256];
44         ostreamstream ms;
45         ms<<"Db rows("<<nrows<<") != compiled("<<NUM_DB_ROWS<<")"<<ends;
46         sendMess((ms.str()).c_str(),dbMFatal,__LINE__,__CLASS__,__METHOD__);
47     }
48     memcpy(elementList,elist,NUM_DB_ROWS*sizeof(int));
49
50     unsigned int timestamp=time(NULL);
51     mgr->setRequestTime(timestamp);
52     if(mgr->fetchDbTable(table)){
53         svtRDOs* thv = (svtRDOs*)table->GetTable();
54         memcpy(previousVals,thv,nrows*sizeof(svtRDOs));
55     }
56
57 #undef __METHOD__
58 };

```

3.3.2.5 void svtRDOsSender::initTags () [virtual]

Definition at line 66 of file svtRDOsSender.cc.

```

66         {
67     /* more than an example -> swap "trg" to your subsys & add to email list*/
68     setEmailTo("porter@bnl.gov");
69     setDomainName("svt");
70
71 }

```

3.3.2.6 bool svtRDOsSender::loadUserControls (const char * *name*, const char * *value*) [virtual]

Definition at line 25 of file svtRDOsSender_i.cc.

```

25                                                                                               {
26 #define __METHOD__ "loadUserControls(name,value)"
27
28 /* more than an example ... swap driftLimit to yours
29 * and duplicate this structure for each selection criteria
30 if(strstr(name,"driftLimit")){
31     driftLimit=atof(value);
32     sendMess("driftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
33     return true;
34 }
35 */
36 if(strstr(name,"vdriftLimit")){
37     vdriftLimit=atof(value);
38     sendMess("vdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
39     return true;
40 }
41 if(strstr(name,"tdriftLimit")){
42     tdriftLimit=atof(value);
43     sendMess("tdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
44     return true;
45 }
46 if(strstr(name,"cdriftLimit")){
47     cdriftLimit=atof(value);
48     sendMess("cdriftLimit set=",value,dbMDebug,__LINE__,__CLASS__,__METHOD__);
49     return true;
50 }
51
52 return false;
53 #undef __METHOD__
54 }

```

3.3.2.7 bool svtRDOsSender::nextLine (ifstream & *from*) [inline]

Definition at line 77 of file svtRDOsSender.hh.

```

77                                                                                               {
78     if(!from.getline(mline,255))return false;
79     return true;
80 }

```

3.3.2.8 bool svtRDOsSender::queryData () [virtual]

Definition at line 91 of file svtRDOsSender.cc.

```

91                                     {
92 #define __METHOD__ "queryData()"
93
94 /*
95  * MORE THAN AN EXAMPLE...
96  * IF Standard SC-Query via "caGet" then,
97  * no need to change this method AT ALL
98  *
99  */
100
101 writeTime = (unsigned int)time(NULL);      //for database write time
102
103 //char systemCmd[1024];
104 ostringstream scmd;
105 scmd<<"caGet "<<queryFile<<" "<<dataFile<<ends;
106
107 if(system((scmd.str()).c_str()))
108     return sendMess(" caGet system call returned error",dbMErr,__LINE__,__CLASS__,__METHOD__);
109
110 return true;
111 #undef __METHOD__
112 };

```

3.3.2.9 char * svtRDOsSender::readAny ()

Definition at line 197 of file svtRDOsSender.cc.

```

197                                     {
198
199 strcpy(tmpLine,mLine);
200 ptr1=tmpLine;
201 ptr2=strtok(ptr1," ");
202 if(!ptr2) return ptr2;
203 ptr2=strtok(NULL," ");
204 return ptr2;
205 }

```

3.3.2.10 bool svtRDOsSender::readData (ifstream & from) [virtual]

Definition at line 110 of file svtRDOsSender_i.cc.

```

110                                     {
111 #define __METHOD__ "readData(ifstream)"
112
113 mreadStatus=true;
114 memset(tempVals,0,NUM_DB_ROWS*sizeof(svtRDOs));
115
116
117
118 char* c=__CLASS__;
119 char* m=__METHOD__;
120
121 for(int i=0;i<NUM_DB_ROWS;i++){
122 if(!nextLine(from) || !readVal(tempVals[i].v1)) readError(__LINE__,c,m);
123 if(!nextLine(from) || !readVal(tempVals[i].v2)) readError(__LINE__,c,m);
124 if(!nextLine(from) || !readVal(tempVals[i].v3)) readError(__LINE__,c,m);
125 if(!nextLine(from) || !readVal(tempVals[i].p6)) readError(__LINE__,c,m);
126 if(!nextLine(from) || !readVal(tempVals[i].mv)) readError(__LINE__,c,m);
127 if(!nextLine(from) || !readVal(tempVals[i].m6)) readError(__LINE__,c,m);
128 if(!nextLine(from) || !readVal(tempVals[i].v1Curr)) readError(__LINE__,c,m);
129 if(!nextLine(from) || !readVal(tempVals[i].v2Curr)) readError(__LINE__,c,m);

```

```

130     if(!nextLine(from) || !readVal(tempVals[i].v3Curr)) readError(__LINE__,c,m);
131     if(!nextLine(from) || !readVal(tempVals[i].p6Curr)) readError(__LINE__,c,m);
132     if(!nextLine(from) || !readVal(tempVals[i].mvCurr)) readError(__LINE__,c,m);
133     if(!nextLine(from) || !readVal(tempVals[i].northTemp)) readError(__LINE__,c,m);
134     if(!nextLine(from) || !readVal(tempVals[i].southTemp)) readError(__LINE__,c,m);
135     if(!nextLine(from) || !readVal(tempVals[i].hvBoardTemp)) readError(__LINE__,c,m);
136     if(!nextLine(from) || !readVal(tempVals[i].optTemp)) readError(__LINE__,c,m);
137     if(!nextLine(from) || !readVal(tempVals[i].lgRingCurr)) readError(__LINE__,c,m);
138     if(!nextLine(from) || !readVal(tempVals[i].rgRingCurr)) readError(__LINE__,c,m);
139     if(!nextLine(from) || !readVal(tempVals[i].lgAnodeCurr)) readError(__LINE__,c,m);
140     if(!nextLine(from) || !readVal(tempVals[i].rgAnodeCurr)) readError(__LINE__,c,m);
141     if(!nextLine(from) || !readVal(tempVals[i].hvVolt)) readError(__LINE__,c,m);
142     if(!nextLine(from) || !readVal(tempVals[i].hvCurr)) readError(__LINE__,c,m);
143 //MPD changes
144     char* tmpval0;
145     if(!nextLine(from)){
146         tmpval0=tempVals[i].hvTrip;
147         if(!readVal(tmpval0))readError(__LINE__,c,m);
148     }
149 //if(!nextLine(from) || !readVal(tempVals[i].hvTrip)) readError(__LINE__,c,m);
150 if(!nextLine(from) || !readVal(tempVals[i].lvFault)) readError(__LINE__,c,m);
151
152     strcpy(tempVals[i].rdo,RDO[i]);
153     tempVals[i].portNum=port[i];
154     if(i<8){ //barrel 1 West
155         tempVals[i].barNum=1;
156         tempVals[i].ladNum=i+1; //sets ladder num = 1-8
157     } else if (i<20) { //barrel 2 West
158         tempVals[i].barNum=2;
159         tempVals[i].ladNum=i-7; //sets ladder num = 1-12
160     } else if (i<36) { //barrel 3 West
161         tempVals[i].barNum=3;
162         tempVals[i].ladNum=i-19; //sets ladder num = 1-16
163     } else if (i<44) { //barrel 1 West
164         tempVals[i].barNum=1;
165         tempVals[i].ladNum=i-35; //sets ladder num = 1-8
166     } else if (i<56) { //barrel 2 West
167         tempVals[i].barNum=2;
168         tempVals[i].ladNum=i-43; //sets ladder num = 1-12
169     } else { //barrel 3 West
170         tempVals[i].barNum=3;
171         tempVals[i].ladNum=i-55; //sets ladder num = 1-16
172     }
173 }
174 }
175
176 /* example format
177 * for(int i=0;i<NUM_DB_ROWS;i++){
178 * if(!nextLine(from) || !readVal(tempVals[i].blah)) readError(1,c,m);
179 *
180 * ....
181 *
182 * }
183 */
184
185 from.close();
186 return true;
187 #undef __METHOD__
188 }

```

3.3.2.11 bool svtRDOsSender::readData (const char *fileName) [virtual]

Definition at line 116 of file svtRDOsSender.cc.

```

116                                     {

```

```

117 #define __METHOD__ "readData(fileName)"
118
119 ifstream from(fileName);
120 if(!from) return sendMess("Cannot open file=",fileName,dbMErr,__LINE__,__CLASS__,__METHOD__);
121
122 return readData(from); // user implemented file read
123 #undef __METHOD__
124 }

```

3.3.2.12 void svtRDOsSender::readError (int *l*, char * *c*, char * *m*) [inline]

Definition at line 82 of file svtRDOsSender.hh.

```

82                                     {
83 mreadStatus=sendMess(" *** Missing Data at ",mline,dbMErr,l,c,m);
84 }

```

3.3.2.13 bool svtRDOsSender::readVal (long long & *value*)

Definition at line 269 of file svtRDOsSender.cc.

```

269                                     {
270
271 if(!readAny())return false;
272 char* store[256];
273 value=strtoll(ptr2,store,10);
274 if(strlen(*store)>0) return false; // value is not a number
275
276 return true;
277 };

```

3.3.2.14 bool svtRDOsSender::readVal (long & *value*)

Definition at line 258 of file svtRDOsSender.cc.

```

258                                     {
259
260 if(!readAny())return false;
261
262 char* store[256];
263 value=strtol(ptr2,store,10);
264 if(strlen(*store)>0) return false; // value is not a number
265
266 return true;
267 };

```

3.3.2.15 bool svtRDOsSender::readVal (int & *value*)

Definition at line 247 of file svtRDOsSender.cc.

```

247                                     {
248
249 if(!readAny()) return false;
250

```

```
251 char* store[256];
252 value=(int)strtol(ptr2,store,10);
253 if(strlen(*store)>0) return false; // value is not a number
254
255 return true;
256 };
```

3.3.2.16 bool svtRDOsSender::readVal (short & value)

Definition at line 236 of file svtRDOsSender.cc.

```
236                                     {
237
238     if(!readAny()) return false;
239
240     char* store[256];
241     value=(short)strtol(ptr2,store,10);
242     if(strlen(*store)>0) return false; // value is not a number
243
244     return true;
245 };
```

3.3.2.17 bool svtRDOsSender::readVal (double & value)

Definition at line 225 of file svtRDOsSender.cc.

```
225                                     {
226
227     if(!readAny())return false;
228
229     char* store[256];
230     value=strtod(ptr2,store);
231     if(strlen(*store)>0) return false; // value is not a number
232
233     return true;
234 };
```

3.3.2.18 bool svtRDOsSender::readVal (float & value)

Definition at line 214 of file svtRDOsSender.cc.

```
214                                     {
215
216     if(!readAny()) return false;
217
218     char* store[256];
219     value=(float)strtod(ptr2,store);
220     if(strlen(*store)>0) return false; // value is not a number
221
222     return true;
223 };
```

3.3.2.19 bool svtRDOsSender::readVal (char *& value)

Definition at line 207 of file svtRDOsSender.cc.

```

207                                     {
208
209     if(!readAny()) return false;
210     strcpy(value,ptr2);
211     return true;
212 }
```

3.3.2.20 bool svtRDOsSender::updateDb (const char *fileName) [virtual]

Definition at line 127 of file svtRDOsSender.cc.

```

127                                     {
128 #define __METHOD__ "updateDb(filename)"
129
130     if(!readData(fileName)) return sendMess(" Read data failed",dbMErr,__LINE__,__CLASS__,__METHOD__);
131
132     int* elements;
133     svtRDOs* vals;
134     int numRows = 0;
135
136     if(writeRequired()){
137
138         numRows=NUM_DB_ROWS;
139         elements=elementList;
140         vals = tempVals;
141
142     } else {
143
144         for(int i=0; i<NUM_DB_ROWS; i++){
145             if(hasChanged(i)){
146                 updateElements[numRows]=elementList[i];
147                 updateVals[numRows] = tempVals[i];
148                 previousVals[i]=tempVals[i];
149                 numRows++;
150             }
151         }
152
153         elements = updateElements;
154         vals      = updateVals;
155     }
156
157     if(numRows==0) return sendMess(" No update required for",mbaseName,dbMDebug,__LINE__,__CLASS__,__MET
158
159     //char mess[256];
160     ostreamstream sn;
161     sn<<"Will Update "<<numRows<<" of "<<NUM_DB_ROWS<<" rows "<<ends;
162     sendMess((sn.str()).c_str(),dbMDebug,__LINE__,__CLASS__,__METHOD__);
163
164     StDbTable* dbTable=node->findTable("svtRDOs");
165     dbTable->SetTable((char*)vals, numRows, elements);
166     mgr->setStoreTime(writeTime);
167
168     if(!mgr->storeDbTable(dbTable)) {
169         addBackLog(writeTime);
170         return sendMess("Store failed ",dbMErr,__LINE__,__CLASS__,__METHOD__);
171     }
172
173     if(numRows==NUM_DB_ROWS)lastFullWrite=writeTime;
174 }
```

```
175     return true;
176     #undef __METHOD__
177 }
```

3.3.3 Member Data Documentation

3.3.3.1 float [svtRDOsSender::cdriftLimit](#) [protected]

Definition at line 39 of file svtRDOsSender.hh.

3.3.3.2 int [svtRDOsSender::elementList](#)[NUM_DB_ROWS] [protected]

Definition at line 24 of file svtRDOsSender.hh.

3.3.3.3 char [svtRDOsSender::mline](#)[256] [protected]

Definition at line 29 of file svtRDOsSender.hh.

3.3.3.4 bool [svtRDOsSender::mreadStatus](#) [protected]

Definition at line 28 of file svtRDOsSender.hh.

3.3.3.5 svtRDOs [svtRDOsSender::previousVals](#)[NUM_DB_ROWS] [protected]

Definition at line 22 of file svtRDOsSender.hh.

3.3.3.6 char* [svtRDOsSender::ptr1](#) [protected]

Definition at line 31 of file svtRDOsSender.hh.

3.3.3.7 char * [svtRDOsSender::ptr2](#) [protected]

Definition at line 31 of file svtRDOsSender.hh.

3.3.3.8 float [svtRDOsSender::tdriftLimit](#) [protected]

Definition at line 38 of file svtRDOsSender.hh.

3.3.3.9 svtRDOs [svtRDOsSender::tempVals](#)[NUM_DB_ROWS] [protected]

Definition at line 23 of file svtRDOsSender.hh.

3.3.3.10 char [svtRDOsSender::tmpline](#)[256] [protected]

Definition at line 30 of file svtRDOsSender.hh.

3.3.3.11 int [svtRDOsSender::updateElements](#)[NUM_DB_ROWS] [protected]

Definition at line 26 of file svtRDOsSender.hh.

3.3.3.12 svtRDOs [svtRDOsSender::updateVals](#)[NUM_DB_ROWS] [protected]

Definition at line 25 of file svtRDOsSender.hh.

3.3.3.13 float [svtRDOsSender::vdriftLimit](#) [protected]

dito

Definition at line 37 of file svtRDOsSender.hh.

The documentation for this class was generated from the following files:

- [svtRDOsSender.hh](#)
- [svtRDOsSender.cc](#)
- [svtRDOsSender_i.cc](#)

Chapter 4

Doxygen_MPD_STAR_ONLINE_DB File Documentation

4.1 svtCoolingDaemon.cc File Reference

```
#include "svtCoolingSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *ldir)

4.1.1 Function Documentation

4.1.1.1 void runSender (const char * ldir)

Definition at line 14 of file svtCoolingDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new svtCoolingSender(ldir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

4.2 svtCoolingSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "svtCoolingSender.hh"
#include "StDbTable.h"
#include "svtCoolingSender_i.cc"
```

Defines

- #define `__CLASS__` "svtCoolingSender"
- #define `__METHOD__` "initTable()"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(fileName)"
- #define `__METHOD__` "updateDb(filename)"

4.2.1 Define Documentation

4.2.1.1 #define `__CLASS__` "svtCoolingSender"

Definition at line 17 of file svtCoolingSender.cc.

4.2.1.2 #define `__METHOD__` "updateDb(filename)"

4.2.1.3 #define `__METHOD__` "readData(fileName)"

4.2.1.4 #define `__METHOD__` "queryData()"

4.2.1.5 #define `__METHOD__` "initDataBase()"

4.2.1.6 #define `__METHOD__` "initTable()"

4.3 svtCoolingSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "svtCooling.h"
```

Classes

- class [svtCoolingSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.3.1 Define Documentation

4.3.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file svtCoolingSender.hh.

4.4 svtCoolingSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

4.4.1 Define Documentation

4.4.1.1 #define `__METHOD__` "readData(ifstream)"

4.4.1.2 #define `__METHOD__` "initQuery()"

4.4.1.3 #define `__METHOD__` "loadUserControls(name,value)"

4.5 svtInterLocksDaemon.cc File Reference

```
#include "svtInterLocksSender.hh"
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *l`dir`)

4.5.1 Function Documentation

4.5.1.1 void runSender (const char * *l`dir`*)

Definition at line 14 of file svtInterLocksDaemon.cc.

```
14         {
15
16     CndDbSender* sender = new svtInterLocksSender(ldir);
17
18     sender->initQuery();
19     for(;;) { //ever...
20         if(sender->hasBackLog())sender->cleanBackLog();
21         if(sender->queryData())sender->updateDb();
22         sleep(sender->sleepTime());
23     }
24
25 };
```

4.6 svtInterLocksSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include "svtInterLocksSender.hh"
#include "StDbTable.h"
#include "svtInterLocksSender_i.cc"
```

Defines

- #define `__CLASS__` "svtInterLocksSender"
- #define `__METHOD__` "initTable()"
- #define `__METHOD__` "initDataBase()"
- #define `__METHOD__` "queryData()"
- #define `__METHOD__` "readData(fileName)"
- #define `__METHOD__` "updateDb(filename)"

4.6.1 Define Documentation

4.6.1.1 #define `__CLASS__` "svtInterLocksSender"

Definition at line 16 of file svtInterLocksSender.cc.

4.6.1.2 #define `__METHOD__` "updateDb(filename)"

4.6.1.3 #define `__METHOD__` "readData(fileName)"

4.6.1.4 #define `__METHOD__` "queryData()"

4.6.1.5 #define `__METHOD__` "initDataBase()"

4.6.1.6 #define `__METHOD__` "initTable()"

4.7 svtInterLocksSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "svtInterLocks.h"
```

Classes

- class [svtInterLocksSender](#)

Defines

- #define [NUM_DB_ROWS](#) 1

4.7.1 Define Documentation

4.7.1.1 #define NUM_DB_ROWS 1

Definition at line 16 of file svtInterLocksSender.hh.

4.8 svtInterLocksSender_i.cc File Reference

Defines

- #define `__METHOD__` "loadUserControls(name,value)"
- #define `__METHOD__` "initQuery()"
- #define `__METHOD__` "readData(ifstream)"

4.8.1 Define Documentation

4.8.1.1 #define `__METHOD__` "readData(ifstream)"

4.8.1.2 #define `__METHOD__` "initQuery()"

4.8.1.3 #define `__METHOD__` "loadUserControls(name,value)"

4.9 svtRDOsDaemon.cc File Reference

```
#include "svtRDOsSender.hh"  
#include <unistd.h>
```

Functions

- void [runSender](#) (const char *lDir)

4.9.1 Function Documentation

4.9.1.1 void runSender (const char * lDir)

Definition at line 14 of file svtRDOsDaemon.cc.

```
14         {  
15  
16     CndDbSender* sender = new svtRDOsSender(lDir);  
17  
18     sender->initQuery();  
19     for(;;) { //ever...  
20         if(sender->hasBackLog())sender->cleanBackLog();  
21         if(sender->queryData())sender->updateDb();  
22         sleep(sender->sleepTime());  
23     }  
24  
25 };
```

4.10 svtRDOsSender.cc File Reference

```
#include <stdlib.h>
#include <unistd.h>
#include <math.h>
#include "svtRDOsSender.hh"
#include "StDbTable.h"
#include "svtRDOsSender_i.cc"
```

Defines

- `#define __CLASS__ "svtRDOsSender"`
- `#define __METHOD__ "initTable()"`
- `#define __METHOD__ "initDataBase()"`
- `#define __METHOD__ "queryData()"`
- `#define __METHOD__ "readData(fileName)"`
- `#define __METHOD__ "updateDb(filename)"`

4.10.1 Define Documentation

4.10.1.1 `#define __CLASS__ "svtRDOsSender"`

Definition at line 17 of file svtRDOsSender.cc.

4.10.1.2 `#define __METHOD__ "updateDb(filename)"`

4.10.1.3 `#define __METHOD__ "readData(fileName)"`

4.10.1.4 `#define __METHOD__ "queryData()"`

4.10.1.5 `#define __METHOD__ "initDataBase()"`

4.10.1.6 `#define __METHOD__ "initTable()"`

4.11 svtRDOsSender.hh File Reference

```
#include "CndDbSender.hh"  
#include "svtRDOs.h"
```

Classes

- class [svtRDOsSender](#)

Defines

- #define [NUM_DB_ROWS](#) 72

4.11.1 Define Documentation

4.11.1.1 #define NUM_DB_ROWS 72

Definition at line 16 of file svtRDOsSender.hh.

Index

- ~svtCoolingSender
 - svtCoolingSender, 6
- ~svtInterLocksSender
 - svtInterLocksSender, 17
- ~svtRDOsSender
 - svtRDOsSender, 27
- __CLASS__
 - svtCoolingSender.cc, 40
 - svtInterLocksSender.cc, 44
 - svtRDOsSender.cc, 48
- __METHOD__
 - svtCoolingSender.cc, 40
 - svtCoolingSender_i.cc, 42
 - svtInterLocksSender.cc, 44
 - svtInterLocksSender_i.cc, 46
 - svtRDOsSender.cc, 48
 - svtRDOsSender_i.cc, 50
- airTempLimit
 - svtCoolingSender, 14
- cdriftLimit
 - svtRDOsSender, 36
- elementList
 - svtCoolingSender, 14
 - svtInterLocksSender, 24
 - svtRDOsSender, 36
- hasChanged
 - svtCoolingSender, 6
 - svtInterLocksSender, 17
 - svtRDOsSender, 27
- initDataBase
 - svtCoolingSender, 7
 - svtInterLocksSender, 17
 - svtRDOsSender, 28
- initQuery
 - svtCoolingSender, 7
 - svtInterLocksSender, 18
 - svtRDOsSender, 28
- initTable
 - svtCoolingSender, 8
 - svtInterLocksSender, 18
 - svtRDOsSender, 29
- initTags
 - svtCoolingSender, 8
 - svtInterLocksSender, 19
 - svtRDOsSender, 29
- loadUserControls
 - svtCoolingSender, 8
 - svtInterLocksSender, 19
 - svtRDOsSender, 30
- mline
 - svtCoolingSender, 14
 - svtInterLocksSender, 24
 - svtRDOsSender, 36
- mreadStatus
 - svtCoolingSender, 14
 - svtInterLocksSender, 24
 - svtRDOsSender, 36
- nextLine
 - svtCoolingSender, 9
 - svtInterLocksSender, 19
 - svtRDOsSender, 30
- NUM_DB_ROWS
 - svtCoolingSender.hh, 41
 - svtInterLocksSender.hh, 45
 - svtRDOsSender.hh, 49
- port
 - svtRDOsSender_i.cc, 50
- previousVals
 - svtCoolingSender, 14
 - svtInterLocksSender, 24
 - svtRDOsSender, 36
- ptr1
 - svtCoolingSender, 14
 - svtInterLocksSender, 24
 - svtRDOsSender, 36
- ptr2
 - svtCoolingSender, 14
 - svtInterLocksSender, 25
 - svtRDOsSender, 36
- queryData
 - svtCoolingSender, 9
 - svtInterLocksSender, 20

- svtRDOsSender, 30
- RDO
 - svtRDOsSender_i.cc, 50
- readAny
 - svtCoolingSender, 10
 - svtInterLocksSender, 20
 - svtRDOsSender, 31
- readData
 - svtCoolingSender, 10, 11
 - svtInterLocksSender, 20, 21
 - svtRDOsSender, 31, 32
- readError
 - svtCoolingSender, 11
 - svtInterLocksSender, 21
 - svtRDOsSender, 33
- readVal
 - svtCoolingSender, 11, 12
 - svtInterLocksSender, 21–23
 - svtRDOsSender, 33, 34
- runSender
 - svtCoolingDaemon.cc, 39
 - svtInterLocksDaemon.cc, 43
 - svtRDOsDaemon.cc, 47
- svtCoolingDaemon.cc, 39
- svtCoolingDaemon.cc
 - runSender, 39
- svtCoolingSender, 5
 - svtCoolingSender, 6
- svtCoolingSender
 - ~svtCoolingSender, 6
 - airTempLimit, 14
 - elementList, 14
 - hasChanged, 6
 - initDataBase, 7
 - initQuery, 7
 - initTable, 8
 - initTags, 8
 - loadUserControls, 8
 - mline, 14
 - mreadStatus, 14
 - nextLine, 9
 - previousVals, 14
 - ptr1, 14
 - ptr2, 14
 - queryData, 9
 - readAny, 10
 - readData, 10, 11
 - readError, 11
 - readVal, 11, 12
 - svtCoolingSender, 6
 - tempVals, 14
 - tmline, 14
 - updateDb, 13
 - updateElements, 14
 - updateVals, 14
 - waterPressLimit, 15
 - waterTempLimit, 15
- svtCoolingSender.cc, 40
- svtCoolingSender.cc
 - __CLASS__, 40
 - __METHOD__, 40
- svtCoolingSender.hh, 41
- svtCoolingSender.hh
 - NUM_DB_ROWS, 41
- svtCoolingSender_i.cc, 42
- svtCoolingSender_i.cc
 - __METHOD__, 42
- svtInterLocksDaemon.cc, 43
- svtInterLocksDaemon.cc
 - runSender, 43
- svtInterLocksSender, 16
 - svtInterLocksSender, 17
- svtInterLocksSender
 - ~svtInterLocksSender, 17
 - elementList, 24
 - hasChanged, 17
 - initDataBase, 17
 - initQuery, 18
 - initTable, 18
 - initTags, 19
 - loadUserControls, 19
 - mline, 24
 - mreadStatus, 24
 - nextLine, 19
 - previousVals, 24
 - ptr1, 24
 - ptr2, 25
 - queryData, 20
 - readAny, 20
 - readData, 20, 21
 - readError, 21
 - readVal, 21–23
 - svtInterLocksSender, 17
 - tempVals, 25
 - tmline, 25
 - updateDb, 23
 - updateElements, 25
 - updateVals, 25
- svtInterLocksSender.cc, 44
- svtInterLocksSender.cc
 - __CLASS__, 44
 - __METHOD__, 44
- svtInterLocksSender.hh, 45
- svtInterLocksSender.hh
 - NUM_DB_ROWS, 45
- svtInterLocksSender_i.cc, 46

- svtInterLocksSender_i.cc
 - __METHOD__, 46
- svtRDOsDaemon.cc, 47
- svtRDOsDaemon.cc
 - runSender, 47
- svtRDOsSender, 26
 - svtRDOsSender, 27
- svtRDOsSender
 - ~svtRDOsSender, 27
 - cdriftLimit, 36
 - elementList, 36
 - hasChanged, 27
 - initDataBase, 28
 - initQuery, 28
 - initTable, 29
 - initTags, 29
 - loadUserControls, 30
 - mline, 36
 - mreadStatus, 36
 - nextLine, 30
 - previousVals, 36
 - ptr1, 36
 - ptr2, 36
 - queryData, 30
 - readAny, 31
 - readData, 31, 32
 - readError, 33
 - readVal, 33, 34
 - svtRDOsSender, 27
 - tdriftLimit, 36
 - tempVals, 36
 - tmpline, 36
 - updateDb, 35
 - updateElements, 36
 - updateVals, 37
 - vdriftLimit, 37
- svtRDOsSender.cc, 48
- svtRDOsSender.cc
 - __CLASS__, 48
 - __METHOD__, 48
- svtRDOsSender.hh, 49
- svtRDOsSender.hh
 - NUM_DB_ROWS, 49
- svtRDOsSender_i.cc, 50
- svtRDOsSender_i.cc
 - __METHOD__, 50
 - port, 50
 - RDO, 50
- svtRDOsSender, 36
- tmpline
 - svtCoolingSender, 14
 - svtInterLocksSender, 25
 - svtRDOsSender, 36
- updateDb
 - svtCoolingSender, 13
 - svtInterLocksSender, 23
 - svtRDOsSender, 35
- updateElements
 - svtCoolingSender, 14
 - svtInterLocksSender, 25
 - svtRDOsSender, 36
- updateVals
 - svtCoolingSender, 14
 - svtInterLocksSender, 25
 - svtRDOsSender, 37
- vdriftLimit
 - svtRDOsSender, 37
- waterPressLimit
 - svtCoolingSender, 15
- waterTempLimit
 - svtCoolingSender, 15
- tdriftLimit
 - svtRDOsSender, 36
- tempVals
 - svtCoolingSender, 14
 - svtInterLocksSender, 25